EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	(Alexander near Klaiber).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L2	91	(David near Dunn).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L3	2331	translation adj lookaside adj buffer	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L4	3274	(translation adj lookaside adj buffer) or TLD	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L5	5214	(translation adj lookaside adj buffer) or TLB	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L6	62	cacheability adj2 (attribute$2 or status or state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L7	6142	PTE	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L8	2462	physical adj page	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L9	119452	miss or mismatch	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19

EAST Search History

L10	2056	(miss or exception) adj handler	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L11	0	walker adj extractor	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L12	360	cache adj flushing	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L13	1198	cache adj flush$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L14	1374	cache adj flush$5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L15	62	cacheability adj2 (attribute$2 or status or state or character$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L16	1	(Alexander near Klaiber).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L17	91	(David near Dunn).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L18	91	L16 OR L17	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:19
L19	62	cacheability adj2 (attribute$2 or status or state or character$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20

EAST Search History

L20	91	L16 OR L17	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L21	0	L19 AND L20	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L22	2056	(miss or exception) adj handler	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L23	33	L22 AND L19	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L24	5214	(translation adj lookaside adj buffer) or TLB	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L25	33	L22 AND L19	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L26	32	L24 AND L25	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L27	659329	page$4	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L28	32	L24 AND L25	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L29	659329	page$4	US-PGPUB;
			USPAT;
			EPO; JPO;
			DERWENT;
			IBM_TDB
				OR	OFF	2005/12/24 12:20

L30	30	L28 and L29	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L31	30	L19 same L24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L32	2164853	compare$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L33	62	cacheability adj2 (attribute$2 or status or state)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L34	2164853	compare$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L35	0	L33 near4 L34	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L36	7	L33 same L34	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L37	7	L33 same L34	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L38	6	L24 and L37	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:20
L39	14151	(roll$4 or recall$4) adj back	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:21

EAST Search History

L40	0	38 and 39	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/12/24 12:21

Home | Login | Logout | Access Information | Ale

IEEE Xplore® RELEASE 2.1

Welcome United States Patent and Trademark Office

Search Session History

BROWSE SEARCH IEEE XPLORE GUIDE

Sat, 24 Dec 2005, 12:28:54 PM EST

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Search Query Display

[redacted] [redacted]

Recent Search Queries

#1	((tlb or (translation lookaside buffer))<in>metadata)
#2	(cache miss<IN>metadata)
#3	((roll back) and coherency<IN>metadata)
#4	((cache flush) and attribute<IN>metadata)
#5	((miss handler) and page<IN>metadata)

[redacted]

Help Contact Us Privac

Indexed by Inspec®

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+translation +lookaside +buffer, +TLB, +cache, +miss, +flush SEARCH

[redacted]

Feedback Report a problem Satisfaction survey

Terms used
translation lookaside buffer TLB cache miss flush rolling or **roll back**

Found **74** of **169,166**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 74 Result page: **1** 2 3 4 next

Relevance scale □ ▂ ▄ ▆ █

1 Tolerating late memory traps in ILP processors █

Xiaogang Qiu, Michel Dubois
May 1999 **ACM SIGARCH Computer Architecture News , Proceedings of the 26th annual international symposium on Computer architecture ISCA '99**, Volume 27 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(100.18 KB) Publisher Site
Additional Information: full citation, abstract, references, citings, index terms

ILP processors can execute a large number of instructions at the same time. Thus it becomes more and more difficult to support traps efficiently. On the other hand a current trend in architecture is to support various memory functions in software rather than hardware, usually by trapping the execution processor on a cache miss, TLB miss or a failed access to a local or remote memory. These late memory traps block the faulting instruction at the top of the active list, backing up the pipeline. Mo ...

2 Recency-based TLB preloading █

Ashley Saulsbury, Fredrik Dahlgren, Per Stenström
May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2
Publisher: ACM Press
Full text available: pdf(651.05 KB)
Additional Information: full citation, abstract, references, citings, index terms

Caching and other latency tolerating techniques have been quite successful in maintaining high memory system performance for general purpose processors. However, TLB misses have become a serious bottleneck as working sets are growing beyond the capacity of TLBs. This work presents one of the first attempts to hide TLB miss latency by using preloading techniques. We present results for traditional next-page TLB miss preloading - an approach shown to cut so ...

3 Multigrain shared memory █

Donald Yeung, John Kubiatowicz, Anant Agarwal
May 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 2
Publisher: ACM Press
Full text available: pdf(369.18 KB)
Additional Information: full citation, abstract, references, index terms,

review

Parallel workstations, each comprising tens of processors based on shared memory, promise cost-effective scalable multiprocessing. This article explores the coupling of such small- to medium-scale shared-memory multiprocessors through software over a local area network to synthesize larger shared-memory systems. We call these systems Distributed Shared-memory MultiProcessors (DSMPs). This article introduces the design of a shared-memory system that uses multiple granularities of sharing, ca ...

Keywords: distributed memory, symmetric multiprocessors, system of systems

4 The use of multithreading for exception handling
Craig B. Zilles, Joel S. Emer, Gurindar S. Sohi
November 1999 **Proceedings of the 32nd annual ACM/IEEE international symposium on Microarchitecture**
Publisher: IEEE Computer Society
Full text available: pdf(1.49 MB) Publisher Site   Additional Information: full citation, abstract, references, citings, index terms

Common hardware exceptions, when implemented by trapping, unnecessarily serialize program execution in dynamically scheduled superscalar processors. To avoid the consequences of trapping the main program thread, multithreaded CPUs can exploit control and data independence by executing the exception handler in a separate hardware context. The main thread doesn't squash instructions after the excepting instruction, conserving fetch bandwidth and allowing execution of instructions inde ...

5 A VLIW architecture for a trace scheduling compiler
Robert P. Colwell, Robert P. Nix, John J. O'Donnell, David B. Papworth, Paul K. Rodman
October 1987 **ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the second international conference on Architectual support for programming languages and operating systems ASPLOS-II**, Volume 15 , 22 , 21 Issue 5 , 10 , 4
Publisher: IEEE Computer Society Press, ACM Press
Full text available: pdf(1.59 MB)   Additional Information: full citation, abstract, references, citings, index terms

Very Long Instruction Word (VLIW) architectures were promised to deliver far more than the factor of two or three that current architectures achieve from overlapped execution. Using a new type of compiler which compacts ordinary sequential code into long instruction words, a VLIW machine was expected to provide from ten to thirty times the performance of a more conventional machine built of the same implementation technology.Multiflow Computer, Inc., has now built a VLIW called the TRACE™< ...

6 Improving the reliability of commodity operating systems
Michael M. Swift, Brian N. Bershad, Henry M. Levy
February 2005 **ACM Transactions on Computer Systems (TOCS)**, Volume 23 Issue 1
Publisher: ACM Press
Full text available: pdf(459.98 KB)   Additional Information: full citation, abstract, references, index terms

Despite decades of research in extensible operating system technology, extensions such as device drivers remain a significant cause of system failures. In Windows XP, for example, drivers account for 85&percent; of recently reported failures.This article describes Nooks, a *reliability subsystem* that seeks to greatly enhance operating system (OS) reliability by isolating the OS from driver failures. The Nooks approach is practical: rather than guaranteeing complete fault tolerance through ...

Keywords: I/O, Recovery, device drivers, protection, virtual memory

7 An in-cache address translation mechanism

D. A. Wood, S. J. Eggers, G. Gibson, M. D. Hill, J. M. Pendleton

June 1986 **ACM SIGARCH Computer Architecture News , Proceedings of the 13th annual international symposium on Computer architecture ISCA '86**, Volume 14 Issue 2

Publisher: IEEE Computer Society Press, ACM Press

Full text available: pdf(770.30 KB) Additional Information: full citation, abstract, references, citings, index terms

In the design of SPUR, a high-performance multiprocessor workstation, the use of large caches and hardware-supported cache consistency suggests a new approach to virtual address translation. By performing translation in each processor's virtually-tagged cache, the need for separate translation lookaside buffers (TLBs) is eliminated. Eliminating the TLB substantially reduces the hardware cost and complexity of the translation mechanism and eliminates the translation consistency problem. Trac ...

8 Cache Memories

Alan Jay Smith

September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3

Publisher: ACM Press

Full text available: pdf(4.61 MB) Additional Information: full citation, references, citings, index terms

9 Translation lookaside buffer consistency: a software approach

D. L. Black, R. F. Rashid, D. B. Golub, C. R. Hill

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the third international conference on Architectural support for programming languages and operating systems ASPLOS-III**, Volume 17 Issue 2

Publisher: ACM Press

Full text available: pdf(1.38 MB) Additional Information: full citation, abstract, references, citings, index terms

We discuss the translation lookaside buffer (TLB) consistency problem for multiprocessors, and introduce the Mach shootdown algorithm for maintaining TLB consistency in software. This algorithm has been implemented on several multiprocessors, and is in regular production use. Performance evaluations establish the basic costs of the algorithm and show that it has minimal impact on application performance. As a result, TLB consistency does not pose an insurmountable obstacle to multiprocessor ...

10 Increasing TLB reach using superpages backed by shadow memory

Mark Swanson, Leigh Stoller, John Carter

April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(1.32 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The amount of memory that can be accessed without causing a TLB fault, the reach of a TLB, is failing to keep pace with the increasingly large working sets of applications. We propose to extend TLB reach via a novel Memory Controller TLB (MTLB) that lets us aggressively create superpages from non-contiguous, unaligned regions of physical memory. This flexibility increases the OS's ability to use superpages on arbitrary application data. The MTLB supports shadow pages, regions of physical address ...

11 Improving the efficiency of UNIX buffer caches
A. Braunstein, M. Riley, J. Wilkes
November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles SOSP '89**, Volume 23 Issue 5
Publisher: ACM Press
Full text available: pdf(1.46 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper reports on the effects of using hardware virtual memory assists in managing file buffer caches in UNIX. A controlled experimental environment was constructed from two systems whose only difference was that one of them (XMF) used the virtual memory hardware to assist file buffer cache search and retrieval. An extensive series of performance characterizations was used to study the effects of varying the buffer cache size (from 3 Megabytes to 70 MB); I\O transfer sizes (from ...

12 Boosting superpage utilization with the shadow memory and the partial-subblock TLB
Cheol Ho Park, JaeWoong Chung, Byeong Hag Seong, YangWoo Roh, Daeyeon Park
May 2000 **Proceedings of the 14th international conference on Supercomputing**
Publisher: ACM Press
Full text available: pdf(798.29 KB) Additional Information: full citation, abstract, references, index terms

While superpage is an efficient solution to increase TLB reach, its limited flexibility for address mapping is still a hard issue. Our proposed mechanism has been developed for taking advantage of two previous approaches which resolve the issue partially: the partial-subblock TLB and the shadow memory. Through integration of them, our mechanism enjoys various benefits inherited from the both sides. By adopting Memory Controller TLB (MTLB) from the shadow memory, it allows superpages to be c ...

13 Options for dynamic address translation in COMAs
Xiaogang Qiu, Michel Dubois
April 1998 **ACM SIGARCH Computer Architecture News , Proceedings of the 25th annual international symposium on Computer architecture ISCA '98**, Volume 26 Issue 3
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(1.37 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

In modern processors, the dynamic translation of virtual addresses to support virtual memory is done before or in parallel with the first-level cache access. As processor technology improves at a rapid pace and the working sets of new applications grow insatiably the latency and bandwidth demands on the TLB (Translation Lookaside Buffer) are getting more and more difficult to meet. The situation is worse in multiprocessor systems, which run larger applications and are plagued by the TLB consiste ...

14 Going the distance for TLB prefetching: an application-driven study
Gokul B. Kandiraju, Anand Sivasubramaniam
May 2002 **ACM SIGARCH Computer Architecture News , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02 , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02**, Volume 30 Issue 2
Publisher: IEEE Computer Society, ACM Press
Full text available: pdf(1.25 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

The importance of the Translation Lookaside Buffer (TLB) on system performance is well known. There have been numerous prior efforts addressing TLB design issues for cutting down access times and lowering miss rates. However, it was only recently that the first exploration [26] on prefetching TLB entries ahead of their need was undertaken and a mechanism called Recency Prefetching was proposed. There is a large body of literature on prefetching for caches, and it is not clear how they can be ada ...

Keywords: application-driven study, memory hierarchy, prefetching, simulation, translation lookaside buffer

15 An architectural perspective on a memory access controller
M. Freeman
June 1987 **Proceedings of the 14th annual international symposium on Computer architecture**
Publisher: ACM Press

Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper a CMOS memory access controller chip is described that provides the basis for achieving high-performance 68020-based (68030-based) systems. This controller matches the speed of the memory system to that of the microprocessor by providing a virtual cache mechanism where address translations are only required when there is a cache miss. This mechanism also facilitates the construction of shared-memory multiprocessor system where the controller manages ...

16 Low-synchronization translation lookaside buffer consistency in large-scale shared-memory multiprocessors
B. Rosenburg
November 1989 **ACM SIGOPS Operating Systems Review , Proceedings of the twelfth ACM symposium on Operating systems principles SOSP '89**, Volume 23 Issue 5
Publisher: ACM Press

Full text available: pdf(1.08 MB) Additional Information: full citation, abstract, references, citings, index terms

Operating systems for most current shared-memory multiprocessors must maintain translation lookaside buffer (TLB) consistency across processors. A processor that changes a shared page table must flush outdated mapping information from its own TLB, and it must force the other processors using the page table to do so as well. Published algorithms for maintaining TLB consistency on some popular commercial multiprocessors incur excessively high synchronization costs. We present an efficient TLB ...

17 High-bandwidth address translation for multiple-issue processors
Todd M. Austin, Gurindar S. Sohi
May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2
Publisher: ACM Press

Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings, index terms

In an effort to push the envelope of system performance, microprocessor designs are continually exploiting higher levels of instruction-level parallelism, resulting in increasing bandwidth demands on the address translation mechanism. Most current microprocessor designs meet this demand with a multi-ported TLB. While this design provides an excellent hit rate at each port, its access latency and area grow very quickly as the number of ports is increased. As bandwidth demands continue to increase ...

18 Eliminating the address translation bottleneck for physical address cache

Tzi-cker Chiueh, Randy H. Katz

September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international conference on Architectural support for programming languages and operating systems ASPLOS-V**, Volume 27 Issue 9

Publisher: ACM Press

Full text available: pdf(1.28 MB) Additional Information: full citation, references, citings, index terms

19 A look at several memory management units, TLB-refill mechanisms, and page table organizations

Bruce L. Jacob, Trevor N. Mudge

October 1998 **ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 32 , 33 Issue 5 , 11

Publisher: ACM Press

Full text available: pdf(1.90 MB) Additional Information: full citation, abstract, references, citings, index terms

Virtual memory is a staple in modem systems, though there is little agreement on how its functionality is to be implemented on either the hardware or software side of the interface. The myriad of design choices and incompatible hardware mechanisms suggests potential performance problems, especially since increasing numbers of systems (even embedded systems) are using memory management. A comparative study of the implementation choices in virtual memory should therefore aid system-level designers ...

20 Supporting reference and dirty bits in SPUR's virtual address cache

D. A. Wood, R. H. Katz

April 1989 **ACM SIGARCH Computer Architecture News , Proceedings of the 16th annual international symposium on Computer architecture ISCA '89**, Volume 17 Issue 3

Publisher: ACM Press

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms

Virtual address caches can provide faster access times than physical address caches, because translation is only required on cache misses. However, because we don't check the translation information on each cache access, maintaining reference and dirty bits is more difficult. In this paper we examine the trade-offs in supporting reference and dirty bits in a virtual address cache. We use measurements from a uniprocessor SPUR prototype to evaluate different alternatives. The prototype's buil ...

Results 1 - 20 of 74 Result page: **1** 2 3 4 next